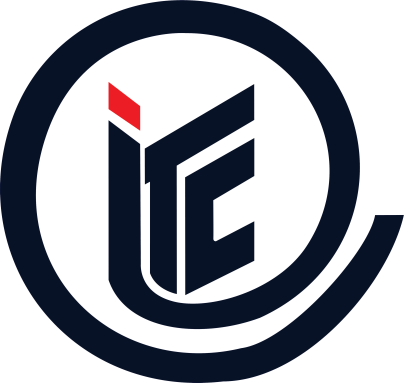
**BỘ LAO ĐỘNG THƯƠNG BINH & XÃ HỘI  
TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THÔNG TIN TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN - ĐIỆN TỬ**



**BÁO CÁO ĐỒ ÁN  
MÔN CÔNG NGHỆ KIỂM THỬ ỨNG DỤNG**

**KIỂM THỬ WEBSITE**

**GIẦY BOM**

**Giảng viên hướng dẫn: Lê Thị Quỳnh Chi**

**Sinh viên thực hiện: Phạm Hồng Huy MSSV: 501210037**

**Huỳnh Phúc Công Anh MSSV: 501210009**

**Đặng Thị Khánh Hiền MSSV: 501210004**

**Tên nhóm học phần: 015\_CNKTUD\_HK1.22-23\_CD21CT1**

**Học kỳ: 1 Năm học: 2022-2023**

**Tp.Hồ Chí Minh, tháng 11 năm 2022**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tp. HCM, ngày….tháng…..năm………  
 (chữ ký và ghi rõ họ tên)

**MỤC LỤC**

[1. GIỚI THIỆU. 1](#_Toc118493669)

[1.1. Mô tả yêu cầu. 1](#_Toc118493670)

[2. NỘI DUNG. 3](#_Toc118493671)

[2.1. Các yêu cầu. 3](#_Toc118493673)

[2.1.1. Yêu cầu B: 3](#_Toc118493674)

[2.1.1.1. Dựa vào trang web lựa chọn, thiết kế database cho phù hợp với dữ liệu trên web. 3](#_Toc118493675)

[2.1.1.2. Làm đầy đủ các quy trình có thể có (giống như thao tác trên web). 4](#_Toc118493676)

[2.1.2. Yêu cầu C: Selenium. 14](#_Toc118493677)

[2.1.2.1. Cài đặt môi trường JDK, ChromeDriver và Selenium Webdriver: 14](#_Toc118493678)

[2.1.2.2. Kiểm thử tự động Unit Test: 18](#_Toc118493679)

[2.1.2.3. Kiểm thử tự động Integration Test: 21](#_Toc118493680)

[2.1.2.3.1. Test case: Chức năng đăng nhập 21](#_Toc118493681)

[2.1.2.3.2. Test case: Chức năng đăng xuất 22](#_Toc118493682)

[2.1.2.3.3. Test case: Chức năng làm trống giỏ hàng khi người dùng đăng xuất 22](#_Toc118493683)

[2.1.2.4. Kiểm thử tự động System: 23](#_Toc118493684)

[2.1.2.4.1. Test case: Chức năng tính lại tổng tiền khi người dùng thay đổi số lượng món hàng trong giỏ 23](#_Toc118493685)

[2.1.3. Yêu cầu D: Kiểm thử hiệu năng đối với 3 thông số sau cho trang web mà nhóm lựa chọn. 24](#_Toc118493686)

[Phần 3: TỔNG KẾT. 32](#_Toc118493687)

[TÀI LIỆU THAM KHẢO 33](#_Toc118493688)

**DANH MỤC HÌNH ẢNH , BIỂU ĐỒ**

[hình 1: mẫu data 1 sản phẩm 3](#_Toc118493704)

[hình 2: mẫu data 1 người dùng 3](#_Toc118493705)

[hình 3: mẫu data 1 đơn hàng 4](#_Toc118493706)

[hình 4: đăng ký thành công 4](#_Toc118493707)

[hình 5: lỗi đăng ký với thông tin email và pass đã tồn tại với test status code 401. 5](#_Toc118493708)

[hình 6: đăng nhập thành công với code 200 6](#_Toc118493709)

[hình 7: đăng nhập thất bại 7](#_Toc118493710)

[hình 8: lấy tất cả sản phẩm và kiểm tra status code 200 8](#_Toc118493711)

[hình 9: lấy ra 1 sản phẩm bằng id và kiểm tra status code 200 8](#_Toc118493712)

[hình 10: lấy toàn bộ đơn hàng thành công lưu mã đơn hàng đàu tiên vào biến đơn hàng 9](#_Toc118493713)

[hình 11: lấy đơn hàng theo mã đơn hàng đã lưu. check status code 200 10](#_Toc118493714)

[hình 12: thêm đơn hàng thành công, trả về số lượng đơn hàng hiện có 11](#_Toc118493715)

[hình 13: xoá đơn hàng bằng mã đơn hàng. code thành công 200 12](#_Toc118493716)

[hình 14: cập nhật đơn hàng thành công 13](#_Toc118493717)

[hình 15: kết quả chạy newman 14](#_Toc118493718)

[hình 16: thiết lập môi trường jdk 15](#_Toc118493719)

[hình 17: cài đặt driver chromium 16](#_Toc118493720)

[hình 18: giải nén và đặt file chromedriver.exe vào resources của project 16](#_Toc118493721)

[hình 19: , chọn nút add để thêm selenium vừa tải 17](#_Toc118493722)

[hình 20: 2.1.2.2.1.test case: kiểm tra trường tên không được phép trống 1 18](#_Toc118493723)

[hình 21: 2.1.2.2.1.test case: kiểm tra trường tên không được phép trống 2 18](#_Toc118493724)

[hình 22: 2.1.2.2.2.test case: kiểm tra trường tên không được chứa số (1) 19](#_Toc118493725)

[hình 23: 2.1.2.2.2.test case: kiểm tra trường tên không được chứa số (2) 19](#_Toc118493726)

[hình 24: test case: kiểm tra trường tên không được chứa kí tự đặc biệt 20](#_Toc118493727)

[hình 25:test case: kiểm tra trường tên không được có khoảng trắng ở đầu 20](#_Toc118493728)

[hình 26: 2.1.2.2.5.test case: kiểm tra trường họ không được phép trống 20](#_Toc118493729)

[hình 27:test case: chức năng đăng nhập 21](#_Toc118493730)

[hình 28: test case: chức năng đăng xuất 22](#_Toc118493731)

[hình 29: test case: chức năng làm trống giỏ hàng khi người dùng đăng xuất 23](#_Toc118493732)

[hình 30: kiểm tra lại số lượng món hàng 23](#_Toc118493733)

[hình 31: 2.1.2.4.1. test case: chức năng tính lại tổng tiền khi người dùng thay đổi số lượng món hàng trong giỏ (1) 23](#_Toc118493734)

[hình 32: 2.1.2.4.1. test case: chức năng tính lại tổng tiền khi người dùng thay đổi số lượng món hàng trong giỏ (2) 24](#_Toc118493735)

[hình 33: khởi động và tạo test plan đặt tên là giaybom trong jmeter. 24](#_Toc118493736)

[hình 34: add jmeter elements 25](#_Toc118493737)

[hình 35:điền thông tin 25](#_Toc118493738)

[hình 36: add listener 26](#_Toc118493739)

[hình 37: kết quả (1) 27](#_Toc118493740)

[hình 38: kết quả (2) 27](#_Toc118493741)

[hình 39: thay đôi số lượng 100 user 28](#_Toc118493742)

[hình 40: kết quả 100 user (1) 28](#_Toc118493743)

[hình 41: kết quả 100 user (2) 29](#_Toc118493744)

[hình 42: tăng lên 1000 user 29](#_Toc118493745)

[hình 43: kết quả 1000 user (1) 30](#_Toc118493746)

[hình 44: kết quả 1000 user (2) 30](#_Toc118493747)

# GIỚI THIỆU.

## Mô tả yêu cầu.

Đồ án kết thúc môn Công nghệ kiểm thử ứng dụng cần đạt được các yêu cầu do giáo viên đặt ra như sau:

**A. Manual Testing**: (Đã làm KTGK)

SV làm tất cả các testcase có thể có cho các chức năng trong trang web đã lựa chọn cho 3 level: unit test, intergration test và system test. Testcase nào bug thì tô màu đỏ.( Lưu ý: tạo 3 sheet riêng cho từng level.)

Do số lượng testcase khá nhiều nên đối với phần Manual testing các thành viên phải phân chia nhau để viết testcase. Trong sheet "phân công" ghi rõ thành viên nào đảm nhận test chức năng nào, v.vv Sử dụng template đính kèm để hoàn thành phần A.

**B. API:** (4đ)

* Dựa vào trang web lựa chọn, thiết kế database cho phù hợp với dữ liệu trên web.
* Làm đầy đủ các quy trình có thể có (giống như thao tác trên web)
* Viết testcase như check status code, check value trả về, kiểm tra giá trị input, v.vv
* Chạy runner + newman

**C. Selenium**: (4đ)

* Những task nào làm auto selenium trong file testcase thì note vào "Yes"
* Quay video chạy các testcase auto -> tải lên google drive và nộp link vào assignment.
* Nén file source code đồ án, tạo folder Selenium và tải toàn bộ lên link github repository của nhóm.

**D. Kiểm thử hiệu năng đối với 3 thông số sau cho trang web mà nhóm lựa chọn:** (1đ)

* Thread group (users): 10/100/1000
* Ram up: 10
* Loop: 1

Sau khi chạy jmeter, dựa vào Listener là View Results Tree/Grapth Results/ Aggregate Report, đưa ra các kết luận, nhận xét về kết quả đạt được cũng như giải pháp đề xuất để nâng cấp hệ thống (nếu có).

# NỘI DUNG.



## Các yêu cầu.

Đường link website: <https://giaybom.com/>

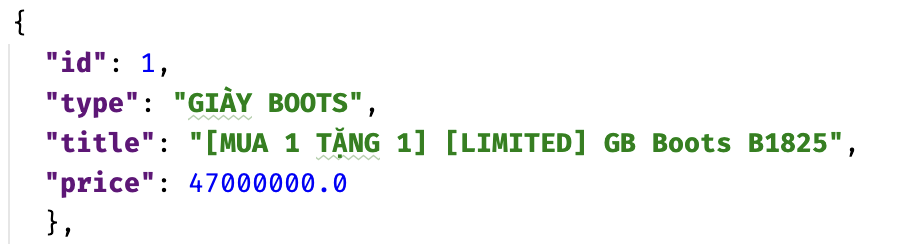
Tiến hành các bước kiểm thử dựa trên trang Web đã chọn

### Yêu cầu B:

#### Dựa vào trang web lựa chọn, thiết kế database cho phù hợp với dữ liệu trên web.

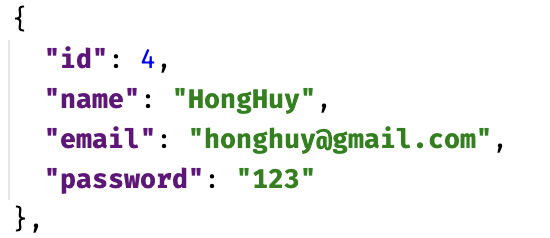
Database thiết kế theo website gồm có:

* Sản phẩm chứa các thông tin như: ID, Loại, Tên, Giá.



Hình 1: Mẫu data 1 sản phẩm

* Người dùng chứa các thông tin: ID, Tên người dùng, Email, Mật khẩu.



Hình 2: Mẫu data 1 người dùng

* Đơn hàng gồm có các thông tin: Mã đơn hàng, mã người dùng, tên người dùng, email, địa chỉ, điện thoại, mã hàng, giá, số lượng, tổng tiền.

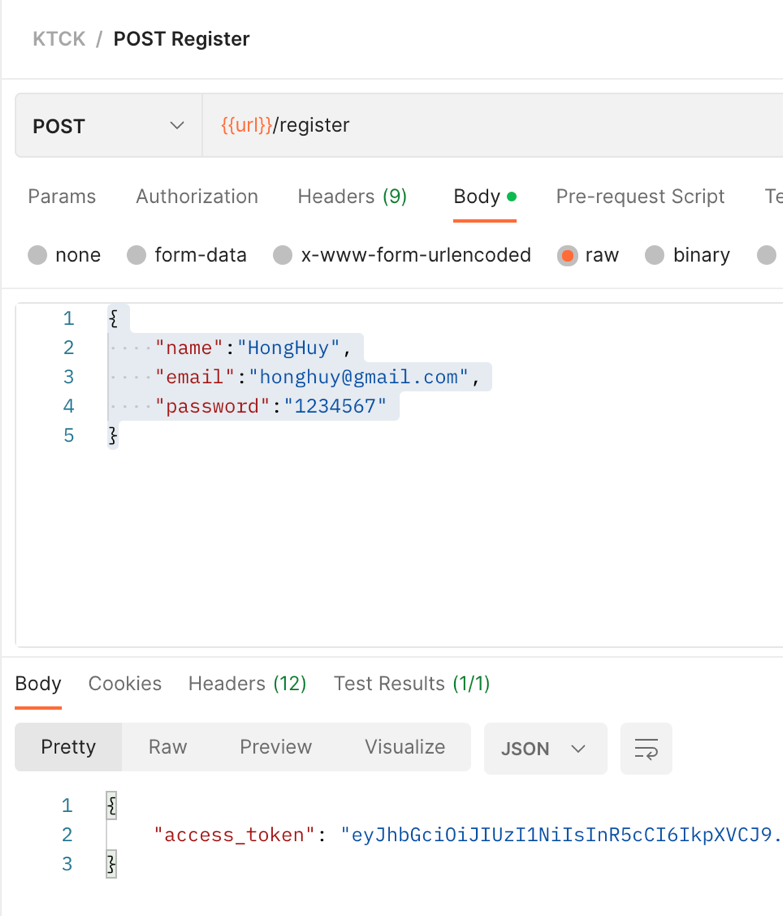
Text

Description automatically generated

Hình 3: Mẫu data 1 đơn hàng

#### Làm đầy đủ các quy trình có thể có (giống như thao tác trên web).

Đối với người dùng, cho phép đăng ký, đăng nhập. Yêu cầu đăng ký không trùng với tài khoản hiện có và email đã được dùng cho tài khoản khác. Kiểm tra đăng ký thành công với Status code 200.



Hình 4: Đăng ký thành công



Hình 5: Lỗi đăng ký với thông tin email và pass đã tồn tại với test Status code 401.

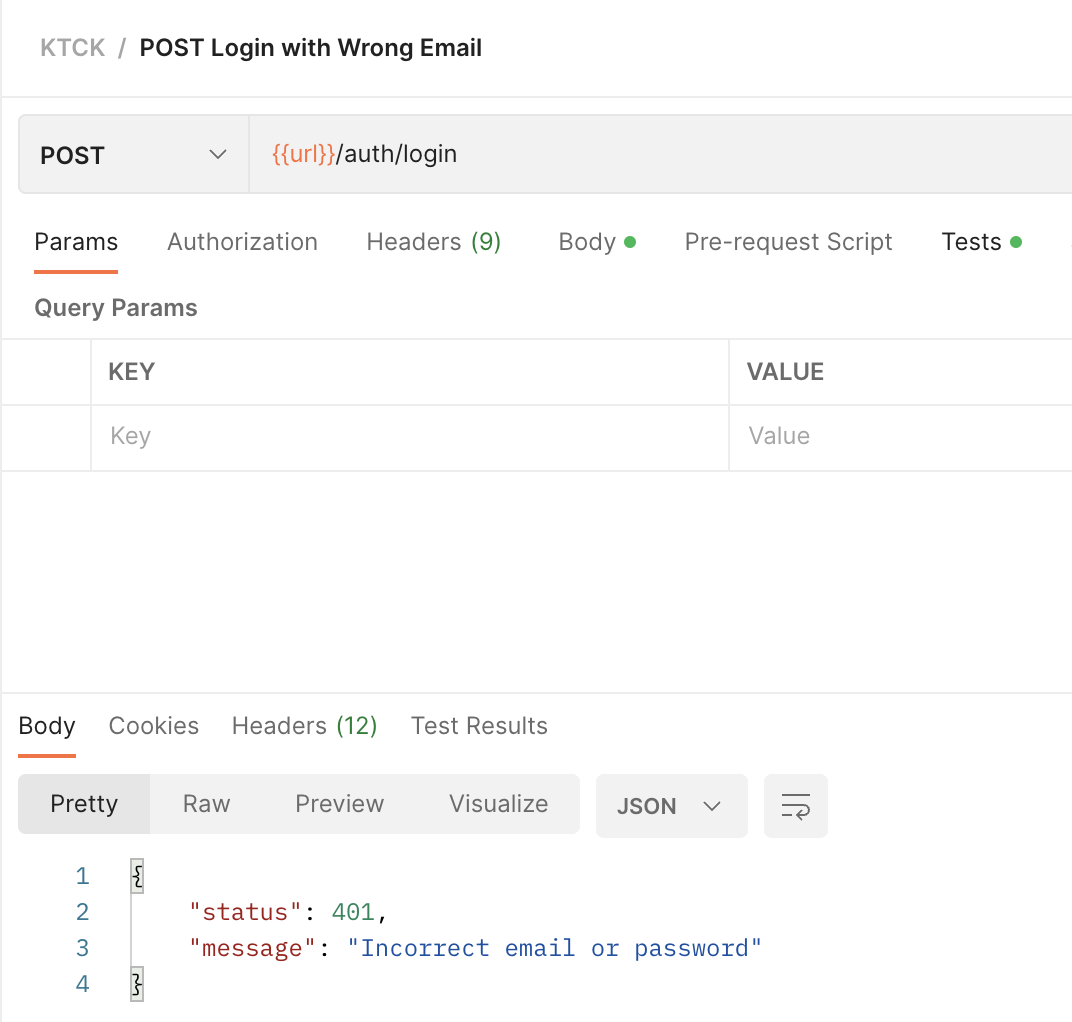
Đăng nhập bằng email và pass thành công kiểm tra Status code 200 sau đó lưu token vào Collection Variable.

Graphical user interface, text, application, email

Description automatically generated

Hình 6: Đăng nhập thành công với code 200

Thất bại với Status code 401.



Hình 7: Đăng nhập thất bại

Đối với sản phẩm. Thiết kế tính năng yêu cầu sản phẩm theo mã hàng hoặc lấy tất cả và Status code trả về 200. Sau đó lấy id của sản phẩm đầu tiên lưu vào Collection Variable.

Graphical user interface, text, application, email

Description automatically generated

Hình 8: Lấy tất cả sản phẩm và kiểm tra Status code 200

Graphical user interface, text, application, email

Description automatically generated

Hình 9: Lấy ra 1 sản phẩm bằng id và kiểm tra Status code 200

Đối với đơn hàng, yêu cầu phải xác thực bằng Bearer Token với giá trị token đã lưu sau khi đăng nhập thành công. Cho phép xem tất cả đơn hàng, thêm, xoá, sửa đơn hàng.

Graphical user interface, text

Description automatically generated

Hình 10: Lấy toàn bộ đơn hàng thành công lưu mã đơn hàng đàu tiên vào biến đơn hàng

Graphical user interface, text, application, email

Description automatically generated

Hình 11: Lấy đơn hàng theo mã đơn hàng đã lưu. Check Status code 200

Graphical user interface, text, application

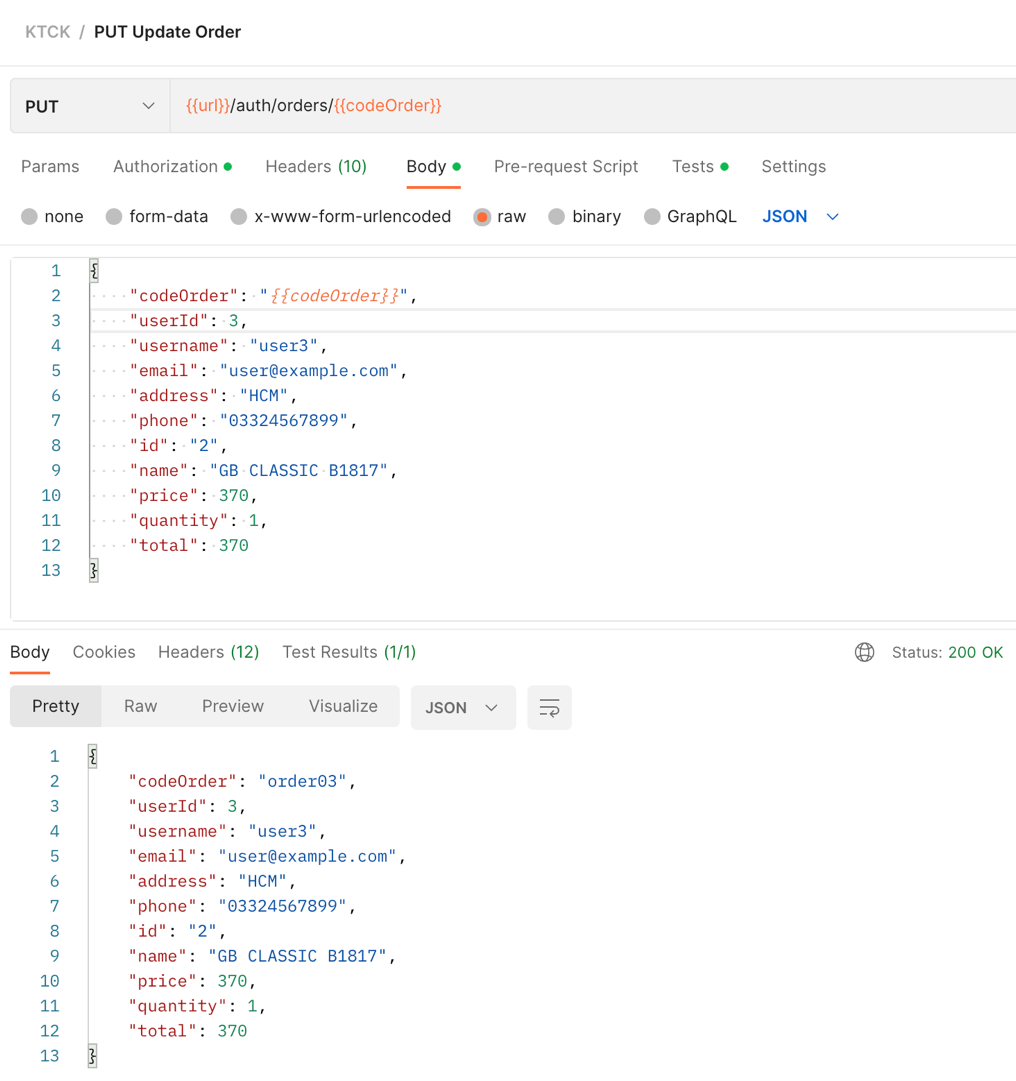
Description automatically generated

Hình 12: Thêm đơn hàng thành công, trả về số lượng đơn hàng hiện có

Graphical user interface, text, application, email

Description automatically generated

Hình 13: Xoá đơn hàng bằng mã đơn hàng. Code thành công 200



Hình 14: Cập nhật đơn hàng thành công

Chạy runner + newman

Text

Description automatically generated

Hình 15: Kết quả chạy Newman

### Yêu cầu C: Selenium.

#### Cài đặt môi trường JDK, ChromeDriver và Selenium Webdriver:

##### Cài đặt môi trường JDK:

Truy cập vào [*Java Downloads | Oracle*](https://www.oracle.com/java/technologies/downloads/) để download môi trường jdk.

Sau đó giải nén và chạy chương trình để Install

##### Cài đặt IDE:

Ở đây nhóm em sử dụng Intellij để viết code nên các bước cài đặt như sau:

* Truy cập vào [*Download IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains*](https://www.jetbrains.com/idea/download/#section=windows) để download phần mềm và cài đặt chúng
* Mở Intellij, mở project mới và thiết lập môi trường tương ứng với jdk đã tải về:

Text

Description automatically generated

Hình 16: Thiết lập môi trường JDK

##### Cài đặt Chrome Driver:

Mở Google Chrome và vào phần Setting kiểm tra version của trình duyệt web đang sử dụng để download driver tương ứng. Đường dẫn: Google Chrome > Dấu ba chấm ở góc phải trên của trình duyệt > Setting > About

Graphical user interface, application

Description automatically generated

Hình 17: Cài đặt Driver Chromium

Ở đây sử dụng phiên bản 107…

Tiếp theo vào [*https://chromedriver.chromium.org/downloads*](https://chromedriver.chromium.org/downloads) để tải xuống chrome driver.

Giải nén và đặt file chromedriver.exe vào resources của project:

Graphical user interface, website

Description automatically generated

Hình 18: Giải nén và đặt file chromedriver.exe vào resources của project

##### Cài đặt Selenium Webdriver:

Trước hết tải Selenium Webdriver về tại đường dẫn: [*[Selenium Java] TÀI NGUYÊN CÀI ĐẶT MÔI TRƯỜNG | Anh Tester*](https://anhtester.com/blog/selenium-java/selenium-java-tai-nguyen-cai-dat-moi-truong) . Đây là file có đuôi là \*.jar và hãy nhớ đường dẫn của nó sau khi tải về.

Trong Intellij, click chuột phải vào project đang làm việc và chọn Open Module Settings.

Tại tab Libraries, chọn nút add để thêm selenium vừa tải về vào bằng đường dẫn của nó:

Text

Description automatically generated

Hình 19: , chọn nút add để thêm selenium vừa tải

Vậy là đã hoàn tất cho việc chuẩn bị môi trường và công cụ để xây dựng các testcases chạy auto rồi nhé! Tiếp theo ta hãy điểm sơ qua một vài ví dụ minh họa của từng hạng mục Manual Testing như Unit test, Integration test, System test.

*Lưu ý:* đây chỉ là một vài minh họa, toàn bộ source nằm tại link github: [*phamhonghuy/DACK-1 (github.com)*](https://github.com/phamhonghuy/DACK-1)

#### Kiểm thử tự động Unit Test:

##### Test case: Kiểm tra trường Tên không được phép trống

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng kí
* Tại trường Tên, không nhập bất cứ gì và điền tất cả trường còn lại
* Nhấn Submit để Đăng kí

Code đã xây dựng:

Text

Description automatically generated

Hình 20: 2.1.2.2.1. Test case: Kiểm tra trường Tên không được phép trống 1

Text

Description automatically generated

Hình 21: 2.1.2.2.1. Test case: Kiểm tra trường Tên không được phép trống 2

##### Test case: Kiểm tra trường Tên không được chứa số

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng kí
* Tại trường Tên, nhập ‘anh123’ và điền tất cả trường còn lại
* Nhấn Submit để Đăng kí

Code đã xây dựng:

Text

Description automatically generated

Hình 22: 2.1.2.2.2. Test case: Kiểm tra trường Tên không được chứa số (1)

Text

Description automatically generated

Hình 23: 2.1.2.2.2. Test case: Kiểm tra trường Tên không được chứa số (2)

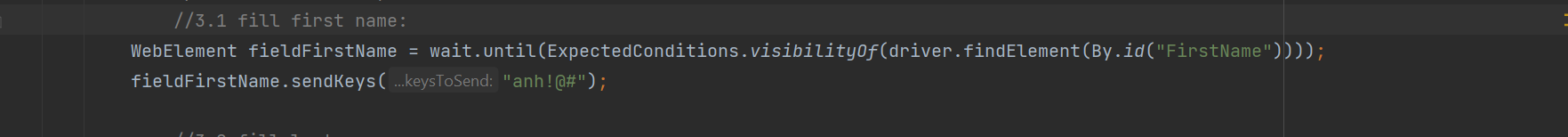
##### Test case: Kiểm tra trường Tên không được chứa kí tự đặc biệt

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng kí
* Tại trường Tên, nhập ‘anh!@#’ và điền tất cả trường còn lại
* Nhấn Submit để Đăng kí

Code đã xây dựng:

Các bước 1, 2 và nhấn Submit giống như trên sẽ không chụp lại. Ta chỉ thay đổi giá trị ở bước nhập dữ liệu cho trường Tên.



Hình 24: Test case: Kiểm tra trường Tên không được chứa kí tự đặc biệt

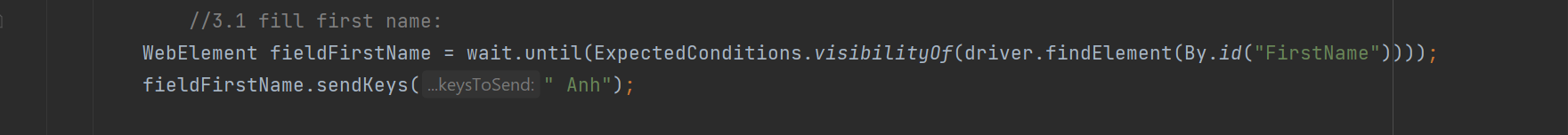
##### Test case: Kiểm tra trường Tên không được có khoảng trắng ở đầu

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng kí
* Tại trường Tên, ‘ anh’ và điền tất cả trường còn lại
* Nhấn Submit để Đăng kí

Code đã xây dựng:

Các bước 1, 2 và nhấn Submit giống như trên sẽ không chụp lại. Ta chỉ thay đổi giá trị ở bước nhập dữ liệu cho trường Tên.



Hình 25: Test case: Kiểm tra trường Tên không được có khoảng trắng ở đầu

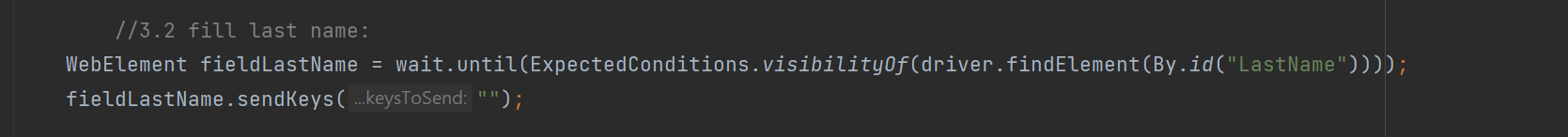
##### Test case: Kiểm tra trường Họ không được phép trống

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng kí
* Tại trường Họ, không nhập bất cứ gì và điền tất cả trường còn lại
* Nhấn Submit để Đăng kí

Code đã xây dựng:

Các bước 1, 2 và nhấn Submit giống như trên sẽ không chụp lại. Ta chỉ thay đổi giá trị ở bước nhập dữ liệu cho trường Họ.



Hình 26: 2.1.2.2.5. Test case: Kiểm tra trường Họ không được phép trống

### Kiểm thử tự động Integration Test:

#### Test case: Chức năng đăng nhập

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng nhập
* Tại trường Email nhập: ‘test6@gmail.com’
* Tại trường password nhập: ‘123456’
* Nhấn Submit để Đăng nhập

Code đã xây dựng:

Text

Description automatically generated

Text

Description automatically generated

Hình 27: Test case: Chức năng đăng nhập

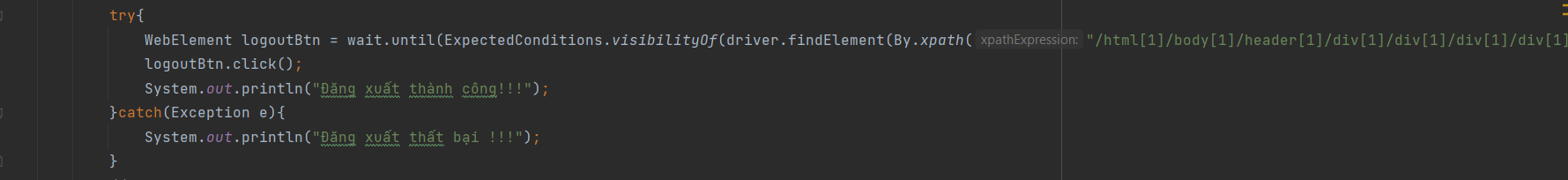
#### Test case: Chức năng đăng xuất

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Nhấn vào nút Đăng nhập
* Tại trường Email nhập: ‘test6@gmail.com’
* Tại trường password nhập: ‘123456’
* Nhấn Submit để Đăng nhập
* Nhấn vào nút Đăng xuất

Code đã xây dựng:

Sau khi đăng nhập theo các bước ở testcase trên thì ta code đoạn đăng xuất như sau:



Hình 28: Test case: Chức năng đăng xuất

#### Test case: Chức năng làm trống giỏ hàng khi người dùng đăng xuất

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Đăng nhập một tài khoản đã đăng kí
* Thêm một vài món hàng vào giỏ
* Đăng xuất và kiểm tra số lượng món hàng trong giỏ sau đó

Code đã xây dựng:

Sau khi đăng nhập theo các bước ở testcase trên thì ta code đoạn đăng thêm hàng và đăng xuất như sau:

Text

Description automatically generated

Hình 29: 2.1.2.3.3. Test case: Chức năng làm trống giỏ hàng khi người dùng đăng xuất

Sau đó kiểm tra lại số lượng món hàng:

A screenshot of a computer

Description automatically generated with medium confidence

Hình 30: kiểm tra lại số lượng món hàng

### Kiểm thử tự động System:

#### Test case: Chức năng tính lại tổng tiền khi người dùng thay đổi số lượng món hàng trong giỏ

Các bước kiểm tra:

* Đi đến trang [*https://giaybom.com/*](https://giaybom.com/)
* Đăng nhập một tài khoản đã đăng kí
* Thêm một vài món hàng vào giỏ
* Thay đổi số lượng của một món hàng trong giỏ
* Nhấn ‘Cập nhật giỏ hàng’ để tính lại giá tiền
* Xác nhận lại đã được thay đổi chưa

Code đã xây dựng cho phần chỉnh sửa số lượng:

Text

Description automatically generated

Hình 31: 2.1.2.4.1. Test case: Chức năng tính lại tổng tiền khi người dùng thay đổi số lượng món hàng trong giỏ (1)

Xác nhận giá tiền sau khi cập nhật:

Text

Description automatically generated

Hình 32: 2.1.2.4.1. Test case: Chức năng tính lại tổng tiền khi người dùng thay đổi số lượng món hàng trong giỏ (2)

### Yêu cầu D: Kiểm thử hiệu năng đối với 3 thông số sau cho trang web mà nhóm lựa chọn.

**Kiểm thử hiệu năng với JMeter**

B1. Khởi động và tạo Test Plan đặt tên là giayBom trong Jmeter.

A picture containing background pattern

Description automatically generated

Hình 33: Khởi động và tạo Test Plan đặt tên là giayBom trong Jmeter.

* Number of Threads: 10 (Số lượng người dùng truy cập trang web: 100)
  + Loop Count: 1 (Thời gian để thực hiện test)
  + Ramp-Up Period: 10s (Trong vòng 10s tạo ra 10 user, tức là trung bình mỗi giây tạo 1 user)

B2: Add JMeter elements

• HTTP request: Click chuột phải Thread Group and chọn: Add -> Sampler -> HTTP Request.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 34: Add JMeter elements

Chúng ta đang thực hiện test trang giayBom.com (https://giaybom.com/) nên chúng ta sẽ điền thông tin như sau

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 35:điền thông tin

B3: Add Listener Click chuột phải vào Test Plan: Add -> Listener -> Graph Results/View Results Tree/Aggregate Report. Ta sẽ có một loại các thư mục tương ứng

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 36: Add Listener

Bước 4: Run Test và xem kết quả Ấn nút Run (Ctrl + R) trên Toolbar để bắt đầu chạy.

Bạn sẽ nhìn thấy kết quả test hiển thị trên Graph. Ở dưới cùng của hình ảnh, bạn sẽ thấy có các số liệu thống kê và được thể hiện bằng màu sắc:

• Đen: Tổng số Sample hiện tại được gửi.

• Màu xanh dương: Trung bình hiện tại của tất cả các Sample được gửi.

• Màu đỏ: Độ lệch chuẩn hiện tại.

• Màu xanh lá cây: Tỷ lệ throughput mà đại diện cho số lượng yêu cầu trên mỗi phút mà máy chủ xử lý

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 37: Kết quả (1)

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 38: Kết quả (2)

Hình ảnh trên chạy tượng trưng cho trang chủ, trang đăng ký, đăng nhập, các trang collection sản phẩm tương ứng.

Sau đó ta tiến hành đổi thông số Thread group (users) thành 100 rồi 1000, để kiểm thử hiệu năng trang web với trường hợp số lượng người dùng truy cập lớn hơn trước.

A picture containing application

Description automatically generated

Hình 39: Thay đôi số lượng 100 user

Kết quả cho thấy trang web chạy khá mượt mà và chưa phát sinh lỗi đối với mức 100 user

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 40: Kết quả 100 User (1)

Chart, line chart

Description automatically generated

Hình 41: Kết quả 100 User (2)

Tiếp tục đổi sang 1000 user như dữ liệu phía dưới

Application

Description automatically generated with low confidence

Hình 42: Tăng lên 1000 User

Kết quả khi thử nghiệm 1000 user đã phát sinh lỗi rất nhiều trên hầu hết các trang được test.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 43: Kết quả 1000 user (1)



Hình 44: Kết quả 1000 user (2)

**Nhận xét kết quả đặt được:**

Trang website chỉ có thể chạy tốt với mức truy cập 100 user, 1 giây 10 user truy cập nhưng khi con số lên đến 1000 thì website gặp tình trạng lỗi quá nhiều, cần khắc phục tình trạng trên nếu không muốn gặp phải tình huống sập website khi lưu lượng người truy cập nhiều đột biến.

Giải phát đề xuất xử lý vấn đề:

* Mua thêm băng thông, thiết lập nhiều server hơn.
* Tối ưu webserver để có thể tiếp nhận nhiều người truy cập hơn.
* Giám sát lưu lượng truy cập để điều chỉnh cân bằng tải.
* Giới hạn lượt truy cập.
* Sử dụng Firewall.

# Phần 3: TỔNG KẾT.

Qua môn học Công nghệ kiểm thử ứng dụng, nhóm em đã có được kỹ năng cần thiết để tiến hành kiểm thử một sản phẩm từ đó cho ra các nhận định, đánh giá về tính ổn định, khả năng thực thi chức năng. Với kiến thức đã học đáp ứng được yêu cầu đẵ đặt ra, và ứng dụng được ngay vào thực tế sử dụng.

Xin chân thành cảm ơn Cô Lê Thị Quỳnh Chi, giáo viên bộ môn Công nghệ kiểm thử ứng dụng. Đã truyền đạt lại rất nhiều kiến thức bổ ích, thực tế về ngành Công nghệ thông tin nói chung và môn Công nghệ kiểm thử ứng dụng nói riêng, đã giúp nhóm em có thêm nhiều kỹ năng cũng như kiến thức bổ ích.

# TÀI LIỆU THAM KHẢO

1. Tài liệu giảng dạy bộ môn Công nghệ kiểm thử ứng dụng.
2. Các nguồn tư liệu từ Internet.