### **Kết quả kiểm thử**

#### *3.3.1.Chức năng đăng ký*

|  |
| --- |
| *public static void register(WebDriver driver, String email, String name, String password, String address, String expectErrMsg) {*  *Utilities.waitForElementVisibility(driver, By.id(EMAIL\_TXT\_ID));*  *Utilities.inputValueAndValidate(driver, By.id(EMAIL\_TXT\_ID), email, email);*  *Utilities.inputValueAndValidate(driver, By.id(NAME\_TXT\_ID), name, name);*  *Utilities.inputValueAndValidate(driver, By.id(PASSWORD\_TXT\_ID), password, password);*  *Utilities.inputValueAndValidate(driver, By.id(ADDRESS\_TXT\_ID), address, address);*  *if (expectErrMsg == Constant.EMPTY\_STRING) {*  *Utilities.clickObscuredElement(driver, By.xpath(REGISTER\_BTN\_XPATH), By.xpath(HomeScreen.USERNAME\_LINK\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *}*  *else {*  *Utilities.clickObscuredElement(driver, By.xpath(REGISTER\_BTN\_XPATH), By.xpath(ERROR\_MSG\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *Utilities.assertTextValue(driver, By.xpath(ERROR\_MSG\_XPATH), expectErrMsg);*  *}*  *}*  *}* |

Đoạn code trên là một phương thức (method) có tên là "register" và có 5 tham số truyền vào là đối tượng WebDriver, email, Tên đăng nhập, mật khẩu, địa chỉ và thông báo lỗi mong đợi

Đoạn mã sử dụng các phương thức khác trong lớp Utilities để chờ đợi sự xuất hiện của phần tử, nhập giá trị vào ô văn bản và xác nhận thông báo lỗi có xuất hiện chính xác hay không.

Thực hiện chạy các ca kiểm thử:

|  |
| --- |
| *public class Register extends Initialization {*  *private static LocalDateTime currentDateTime;*  *private static DateTimeFormatter formatter;*  *private static String formattedDateTime;*    *@BeforeClass()*  *public void setUpClass() throws Exception {*  *driver = RegisterScreen.openScreen(browser);*  *}*  *@BeforeMethod()*  *public void setUpMethod(Method method) throws Exception {*  *Utilities.testID = method.getName();*  *if (Utilities.checkElementVisible(driver, By.xpath(HomeScreen.USERNAME\_LINK\_XPATH))) {*  *HomeScreen.logout(driver);*  *}*  *Utilities.clickObscuredElement(driver, By.className(HomeScreen.REGISTER\_LINK\_CLASS), By.xpath(RegisterScreen.REGISTER\_BTN\_XPATH),*  *Constant.WAIT\_ELEMENT\_EXIST);*  *// Get current time*  *currentDateTime = LocalDateTime.now();*  *formatter = DateTimeFormatter.ofPattern("yyyyMMddHHmmss");*  *formattedDateTime = currentDateTime.format(formatter);*  *}*  *@Test()*  *public void TC\_01\_01() throws IOException {*  *RegisterScreen.register(driver, Constant.BASE\_EMAIL.replace(Constant.TEST\_TEXT, formattedDateTime),*  *Constant.BASE\_NAME, Constant.BASE\_PASSWORD, Constant.BASE\_ADDRESS, Constant.EMPTY\_STRING);*  *}*  *@Test()*  *public void TC\_01\_02() throws IOException {*  *RegisterScreen.register(driver, Constant.EMPTY\_STRING, Constant.BASE\_NAME, Constant.BASE\_PASSWORD,*  *Constant.BASE\_ADDRESS, RegisterScreen.EMPTY\_EMAIL\_MSG);*  *}*    *@Test()*  *public void TC\_01\_03() throws IOException {*  *RegisterScreen.register(driver, Constant.FULL\_SPACE\_STRING, Constant.BASE\_NAME, Constant.BASE\_PASSWORD,*  *Constant.BASE\_ADDRESS, RegisterScreen.EMPTY\_EMAIL\_MSG);*  *}*  *….* |

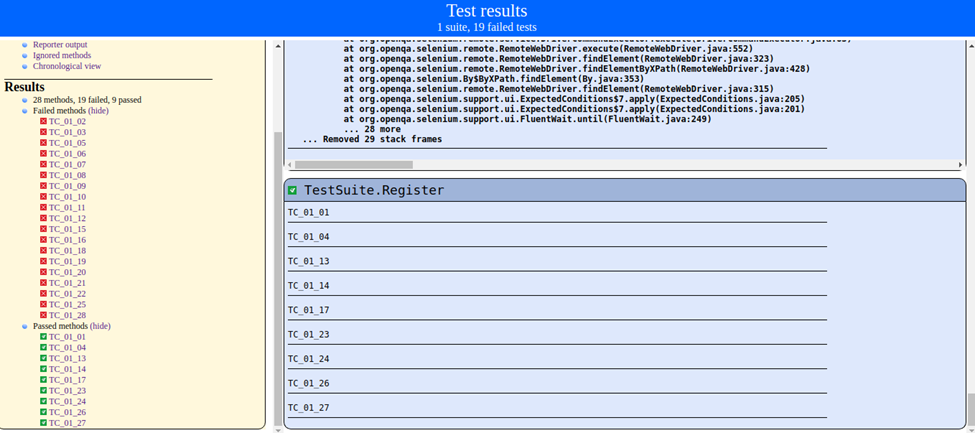
**Kết quả:**

Test suite output thường được sử dụng để báo cáo kết quả của các bộ kiểm tra (test suite) được thực hiện. Khi chạy một test suite, các đầu ra có thể được phân loại thành passed (đã qua), failed (không qua), hoặc skipped (bỏ qua). Các đầu ra này được sắp xếp trong báo cáo kết quả để người dùng có thể biết được trạng thái của việc kiểm tra.

· Passed: Đầu ra cho biết rằng tất cả các test case đã chạy thành công và không có lỗi nào được phát hiện.

· Failed: Đầu ra cho biết rằng một hoặc nhiều test case đã thất bại vì một lý do nào đó.

· Skipped: Đầu ra cho biết rằng một hoặc nhiều test case đã bị bỏ qua vì chúng không phù hợp với điều kiện nào đó.



#### 3.3.2. Chức năng đăng nhập

|  |
| --- |
| *public static void login(WebDriver driver, String email, String password, String expectErrMsg) {*  *Utilities.waitForElementVisibility(driver, By.id(EMAIL\_TXT\_ID));*  *Utilities.inputValueAndValidate(driver, By.id(EMAIL\_TXT\_ID), email, email);*  *Utilities.inputValueAndValidate(driver, By.id(PASSWORD\_TXT\_ID), password, password);*  *if (expectErrMsg == "") {*  *Utilities.clickObscuredElement(driver, By.xpath(LOGIN\_BTN\_XPATH), By.xpath(HomeScreen.USERNAME\_LINK\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *}*  *else if (expectErrMsg == EMPTY\_EMAIL\_MSG || expectErrMsg == EMPTY\_PASSWORD\_MSG) {*  *Utilities.clickObscuredElement(driver, By.xpath(LOGIN\_BTN\_XPATH), By.xpath(ERROR\_MSG\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *Utilities.assertTextValue(driver, By.xpath(ERROR\_MSG\_XPATH), expectErrMsg);*  *}*  *else {*  *Utilities.clickObscuredElement(driver, By.xpath(LOGIN\_BTN\_XPATH), By.xpath(ERROR\_MSG\_XPATH2), Constant.WAIT\_ELEMENT\_EXIST);*  *Utilities.assertTextValue(driver, By.xpath(ERROR\_MSG\_XPATH2), expectErrMsg);*  *}*  *}* |

sử dụng hàm Utilities.waitForElementVisibility để chờ cho phần tử có ID là EMAIL\_TXT\_ID (có thể là một phần tử input email) xuất hiện trên trang web.

Tiếp theo, hàm sử dụng hàm Utilities.inputValueAndValidate để điền giá trị email vào trường input có ID là EMAIL\_TXT\_ID và kiểm tra xem giá trị đã được điền chính xác hay không. Tương tự đối với trường Password.

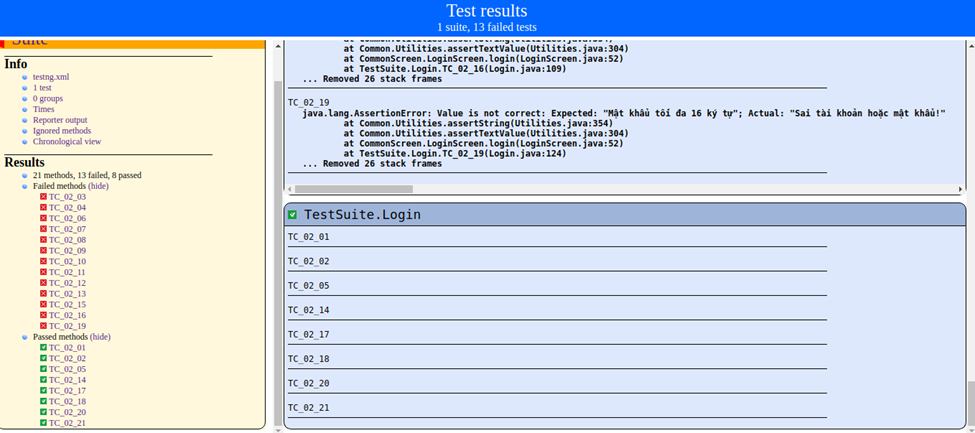
Tiếp theo là một loạt các điều kiện kiểm tra để xác định hành động cần thực hiện dựa trên expectErrMsg.

* Nếu expectErrMsg trống (rỗng), tức là không mong đợi bất kỳ thông điệp lỗi nào, hàm sẽ nhấp vào nút Đăng nhập (dựa trên XPath) và sau đó kiểm tra một phần tử nào đó xuất hiện để đảm bảo rằng đăng nhập thành công (chẳng hạn như nút tên người dùng).
* Trong trường hợp expectErrMsg chứa các thông điệp lỗi cụ thể như EMPTY\_EMAIL\_MSG hoặc EMPTY\_PASSWORD\_MSG, hàm sẽ nhấp vào nút Đăng nhập và kiểm tra xem thông điệp lỗi tương ứng xuất hiện hay không.
* Trong trường hợp expectErrMsg không rỗng và không trùng với hai thông điệp lỗi trên, hàm sẽ kiểm tra một thông điệp lỗi khác (có thể dựa trên XPath) và so sánh với expectErrMsg.

|  |
| --- |
| public class Login extends Initialization {  @BeforeClass()  public void setUpClass() throws Exception {  driver = LoginScreen.openScreen(browser);  }  @BeforeMethod()  public void setUpMethod(Method method) throws Exception {  Utilities.testID = method.getName();  if (Utilities.checkElementVisible(driver, By.xpath(HomeScreen.USERNAME\_LINK\_XPATH))) {  HomeScreen.logout(driver);  }  Utilities.clickObscuredElement(driver, By.className(HomeScreen.LOGIN\_LINK\_CLASS), By.xpath(LoginScreen.LOGIN\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);  }  @Test()  public void TC\_02\_01() throws IOException {  LoginScreen.login(driver, Constant.BASE\_EMAIL, Constant.BASE\_PASSWORD, Constant.EMPTY\_STRING);  }  @Test()  public void TC\_02\_02() throws IOException {  LoginScreen.login(driver, Constant.BASE\_EMAIL\_UPPER, Constant.BASE\_PASSWORD, Constant.EMPTY\_STRING);  }  … |

Lớp này để xác minh tính chính xác của tính năng đăng nhập của một ứng dụng bằng cách chạy nhiều trường hợp kiểm tra cho các kịch bản khác nhau.

**Kết quả**:

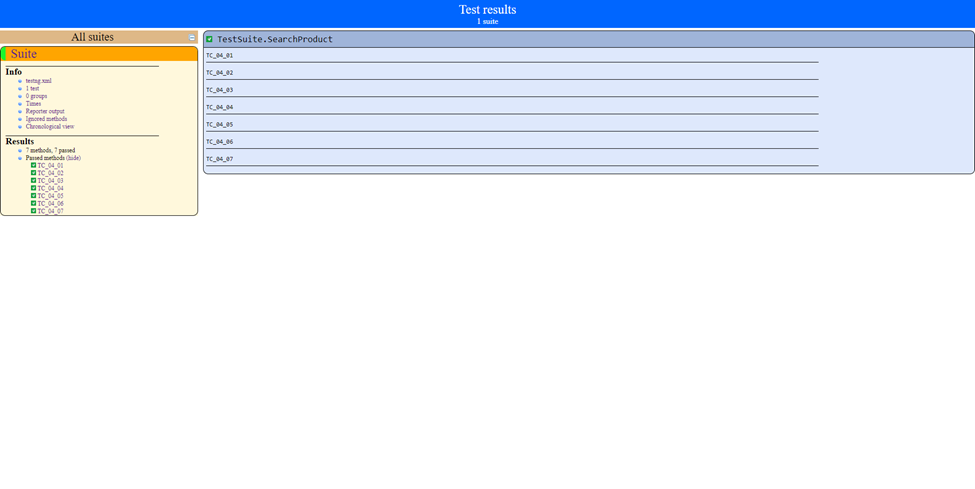


#### 3.3.3. Tìm kiếm sản phẩm:

|  |
| --- |
| public class SearchProduct extends Initialization {  @BeforeClass()  public void setUpClass() throws Exception {  driver = HomeScreen.openScreenWithoutLogin(browser);  }  @BeforeMethod()  public void setUpMethod(Method method) throws Exception {  Utilities.testID = method.getName();  Utilities.clickObscuredElement(driver, By.className(HomeScreen.SEARCH\_ICON\_CLASS), By.xpath(HomeScreen.SEARCH\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);  }  @Test()  public void TC\_04\_01() throws IOException {  HomeScreen.searchProduct(driver, Constant.EMPTY\_STRING, true);  }  @Test()  public void TC\_04\_02() throws IOException {  HomeScreen.searchProduct(driver, Constant.FULL\_SPACE\_STRING, true);  }  @Test()  public void TC\_04\_03() throws IOException {  HomeScreen.searchProduct(driver, Constant.VALID\_KEYWORD1, true);  } |

Mỗi phương thức kiểm thử (VD: TC\_04\_01, TC\_04\_02, TC\_04\_03) đều thực hiện một loạt thao tác tìm kiếm sản phẩm trên trang web thông qua phương thức searchProduct được gọi từ HomeScreen và với các tham số khác nhau (chẳng hạn như Constant.EMPTY\_STRING, Constant.FULL\_SPACE\_STRING, Constant.VALID\_KEYWORD1).

Kết quả cho các ca kiểm thử:



#### **3.3.4. Thay đổi mật khẩu:**

|  |
| --- |
| public static void changePassword(WebDriver driver, String oldPassword, String newPassword, String newPasswordConfirm, String expectErrMsg) {  Utilities.inputValueAndValidate(driver, By.id(OLD\_PASSWORD\_TXT\_ID), oldPassword, oldPassword);  Utilities.inputValueAndValidate(driver, By.id(NEW\_PASSWORD\_TXT\_ID), newPassword, newPassword);  Utilities.inputValueAndValidate(driver, By.id(NEW\_PASSWORD\_CONFIRM\_TXT\_ID), newPasswordConfirm, newPasswordConfirm);    if (expectErrMsg.equals(Constant.EMPTY\_STRING)) {  Utilities.clickObscuredElement(driver, By.xpath(UPDATE\_BTN\_XPATH), By.xpath(SUCCESS\_MSG\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);  Utilities.assertTextValue(driver, By.xpath(SUCCESS\_MSG\_XPATH), CHANGE\_PASSWORD\_SUCCESS\_MSG);  }  else if (expectErrMsg.equals(SAME\_OLD\_PASSWORD\_MSG) || expectErrMsg.equals(INCORRECT\_PASSWORD\_MSG) || expectErrMsg.equals(UNMATCHING\_PASSWORD\_MSG)) {  Utilities.clickObscuredElement(driver, By.xpath(UPDATE\_BTN\_XPATH), By.xpath(UNSUCCESS\_MSG\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);  Utilities.assertTextValue(driver, By.xpath(UNSUCCESS\_MSG\_XPATH), expectErrMsg);  }  else {  Utilities.clickObscuredElement(driver, By.xpath(UPDATE\_BTN\_XPATH), By.xpath(ERROR\_MSG\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);  Utilities.assertTextValue(driver, By.xpath(ERROR\_MSG\_XPATH), expectErrMsg);  }  } |

Hàm này thực hiện việc thay đổi mật khẩu trên trang web đã đăng nhập trước đó. Nó nhận các tham số sau:

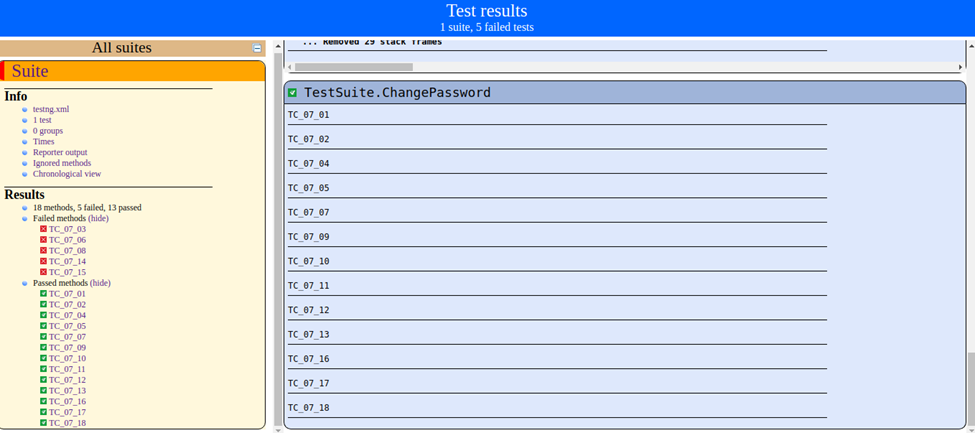
* oldPassword: Mật khẩu cũ.
* newPassword: Mật khẩu mới.
* newPasswordConfirm: Xác nhận mật khẩu mới.
* expectErrMsg: Chuỗi thông điệp lỗi dự kiến hoặc trống nếu không mong đợi thông điệp lỗi nào.

Hàm này thực hiện việc điền giá trị của mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới vào các trường input tương ứng trên trang web.

**Thực hiện các ca kiểm thử:**

|  |
| --- |
| *@BeforeMethod()*  *public void setUpMethod(Method method) throws Exception{*  *Utilities.testID = method.getName();*  *if (Utilities.checkElementVisible(driver, By.className(HomeScreen.LOGIN\_LINK\_CLASS))) {*  *Utilities.clickObscuredElement(driver, By.className(HomeScreen.LOGIN\_LINK\_CLASS), By.xpath(LoginScreen.LOGIN\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *LoginScreen.login(driver, Constant.BASE\_EMAIL, newPassword, Constant.EMPTY\_STRING);*  *Utilities.clickObscuredElement(driver, HomeScreen.USERNAME\_LINK\_XPATH, HomeScreen.CHANGE\_PASSWORD\_LINK\_XPATH, Constant.WAIT\_ELEMENT\_EXIST);*  *Utilities.clickObscuredElement(driver, By.xpath(HomeScreen.CHANGE\_PASSWORD\_LINK\_XPATH), By.xpath(ChangePasswordScreen.UPDATE\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *ChangePasswordScreen.changePassword(driver, newPassword, Constant.BASE\_PASSWORD, Constant.BASE\_PASSWORD, Constant.EMPTY\_STRING);*  *Utilities.clickObscuredElement(driver, By.className(HomeScreen.LOGIN\_LINK\_CLASS), By.xpath(LoginScreen.LOGIN\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *LoginScreen.login(driver, Constant.BASE\_EMAIL, Constant.BASE\_PASSWORD, Constant.EMPTY\_STRING);*  *}*  *Utilities.clickObscuredElement(driver, HomeScreen.USERNAME\_LINK\_XPATH, HomeScreen.CHANGE\_PASSWORD\_LINK\_XPATH, Constant.WAIT\_ELEMENT\_EXIST);*  *Utilities.clickObscuredElement(driver, By.xpath(HomeScreen.CHANGE\_PASSWORD\_LINK\_XPATH), By.xpath(ChangePasswordScreen.UPDATE\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *}*    *@Test()*  *public void TC\_07\_01() throws IOException{*  *newPassword = Constant.BASE\_PASSWORD2;*  *ChangePasswordScreen.changePassword(driver, Constant.BASE\_PASSWORD, newPassword, newPassword, Constant.EMPTY\_STRING);*  *}*    *@Test()*  *public void TC\_07\_02() throws IOException{*  *newPassword = Constant.BASE\_PASSWORD2;*  *ChangePasswordScreen.changePassword(driver, Constant.EMPTY\_STRING, newPassword, newPassword, ChangePasswordScreen.EMPTY\_OLD\_PASSWORD\_MSG);*  *}*    *@Test()*  *public void TC\_07\_03() throws IOException{*  *newPassword = Constant.BASE\_PASSWORD2;*  *ChangePasswordScreen.changePassword(driver, Constant.BASE\_PASSWORD1, newPassword, newPassword, ChangePasswordScreen.MIN\_PASSWORD\_MSG);*  *}*    *@Test()*  *public void TC\_07\_04() throws IOException{*  *newPassword = Constant.BASE\_PASSWORD2;*  *ChangePasswordScreen.changePassword(driver, Constant.BASE\_PASSWORD, newPassword, newPassword, Constant.EMPTY\_STRING);*  *}*  *…* |

Kết quả ca kiểm thử:



**3.4.5. Đăng xuất**

|  |
| --- |
| *public class Logout extends Initialization {*  *@BeforeClass()*  *public void setUpClass() throws Exception {*  *driver = HomeScreen.openScreen(browser);*  *}*  *@BeforeMethod()*  *public void setUpMethod(Method method) throws Exception {*  *Utilities.testID = method.getName();*  *}*  *@Test()*  *public void TC\_03\_01() throws IOException {*  *HomeScreen.logout(driver);*  *}*  *}* |

**Thực hiện chạy các ca kiểm thử:**

|  |
| --- |
| *@BeforeMethod()*  *public void setUpMethod(Method method) throws Exception {*  *Utilities.testID = method.getName();*  *if (Utilities.checkElementVisible(driver, By.xpath(HomeScreen.USERNAME\_LINK\_XPATH))) {*  *HomeScreen.logout(driver);*  *}*  *Utilities.clickObscuredElement(driver, By.className(HomeScreen.LOGIN\_LINK\_CLASS), By.xpath(LoginScreen.LOGIN\_BTN\_XPATH), Constant.WAIT\_ELEMENT\_EXIST);*  *}*  *@Test()*  *public void TC\_02\_01() throws IOException {*  *LoginScreen.login(driver, Constant.BASE\_EMAIL, Constant.BASE\_PASSWORD, Constant.EMPTY\_STRING);*  *}*  *@Test()*  *public void TC\_02\_02() throws IOException {*  *LoginScreen.login(driver, Constant.BASE\_EMAIL\_UPPER, Constant.BASE\_PASSWORD, Constant.EMPTY\_STRING);*  *}*    *@Test()*  *public void TC\_02\_03() throws IOException {*  *LoginScreen.login(driver, Constant.EMPTY\_STRING, Constant.BASE\_PASSWORD, LoginScreen.EMPTY\_EMAIL\_MSG);*  *}*    *@Test()*  *public void TC\_02\_04() throws IOException {*  *LoginScreen.login(driver, Constant.FULL\_SPACE\_STRING, Constant.BASE\_PASSWORD, LoginScreen.EMPTY\_EMAIL\_MSG);*  *}*  *…* |

Kết quả:

