**Test Cases -** [**https://www.saucedemo.com/**](https://www.saucedemo.com/)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID: TC01** | |  | | |
| **Functionality: Login** | |  | | |
| **Test Title: Verify login with valid username and password** | |  | | |
| **Description: Test the login page by entering the valid credentials** | |  | | |
|  | | | | |
| **Pre-conditions: The user with a valid username and password exists in the database** | | | | |
|  | | | | |
| **Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1. **Go to the website** [**https://www.saucedemo.com/**](https://www.saucedemo.com/) 2. **Enter the valid username and password in the username and password fields** 3. **Click the “LOGIN” button** | **Username: “standard\_user”**  **Password:**  **“secret\_sauce”** | **Successful login** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID: TC02** | |  | | |
| **Functionality: Login** | |  | | |
| **Test Title: Verify login with empty username** | |  | | |
| **Description: Test the login page by not entering the username** | |  | | |
|  | | | | |
| **Pre-conditions: The user with a valid username and password exists in the database** | | | | |
|  | | | | |
| **Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1. **Go to the website** [**https://www.saucedemo.com/**](https://www.saucedemo.com/) 2. **Do not enter the username; leave it empty** 3. **Enter the password** 4. **Click the “LOGIN” button** | **Password:**  **“secret\_sauce”** | **Error message; user not logged in** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID: TC03** | |  | | |
| **Functionality: Login** | |  | | |
| **Test Title: Verify login with empty password** | |  | | |
| **Description: Test the login page by not entering the password** | |  | | |
|  | | | | |
| **Pre-conditions: The user with a valid username and password exists in the database** | | | | |
|  | | | | |
| **Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1. **Go to the website** [**https://www.saucedemo.com/**](https://www.saucedemo.com/) 2. **Enter the username** 3. **Do not enter the password; leave it empty** 4. **Click the “LOGIN” button** | **Username: “standard\_user”** | **Error message; user not logged in** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID: TC04** | |  | | |
| **Functionality: Login** | |  | | |
| **Test Title: Verify login with invalid username and password** | |  | | |
| **Description: Test the login page by entering the invalid credentials** | |  | | |
|  | | | | |
| **Pre-conditions:** | | | | |
|  | | | | |
| **Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1. **Go to the website** [**https://www.saucedemo.com/**](https://www.saucedemo.com/) 2. **Enter the invalid username and password in the username and password fields** 3. **Click the “LOGIN” button** | **Username: “username”**  **Password:**  **“password”** | **Error message; user not logged in** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID: TC05** | |  | | |
| **Functionality: Shopping cart** | |  | | |
| **Test Title: Verify the shopping cart stores the added item information** | |  | | |
| **Description: Test the shopping cart by adding the item to the cart and checking whether the cart stores information about the item quantity, name, description and price** | |  | | |
|  | | | | |
| **Pre-conditions: There must be an item existing in the database** | | | | |
|  | | | | |
| **Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1. **Go to the website** [**https://www.saucedemo.com/**](https://www.saucedemo.com/) 2. **Login to the website through the Login page** 3. **Add an item to the cart by clicking the “ADD TO CART” button** 4. **Go to the shopping cart by clicking the cart icon** | **Item to be added to the cart “Sauce Labs Backpack”** | **The cart displays the item quantity, name, price and description** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID: TC06** | |  | | |
| **Functionality: Checkout** | |  | | |
| **Test Title: Verify the personal information form on the checkout page cannot be submitted if the required fields are empty** | |  | | |
| **Description: Test the personal information form by leaving the fields empty** | |  | | |
|  | | | | |
| **Pre-conditions: There must be the item in the cart** | | | | |
|  | | | | |
| **Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| 1. **Go to the website** [**https://www.saucedemo.com/**](https://www.saucedemo.com/) 2. **Login to the website through the Login page** 3. **Add an item to the cart by clicking the “ADD TO CART” button** 4. **Go to the shopping cart by clicking the cart icon** 5. **Click on the “CHECKOUT” button; it will redirect to the personal information form** 6. **Fill up the form** 7. **Click the “CONTINUE” button** 8. **Click the “FINISH” button** | **Item to be added to the cart “Sauce Labs Backpack”**  **Information to be used to fill-up the form**  **First name “James”**  **Last name “Smith”**  **Zip/Postal Code “2016”** | **The thank you message is displayed** |  |  |

**Project**

**Extended tests**

**LoginPageExtended**

package com.planittesting.swaglabs.model.pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class LoginPageExtended extends BasePage {

  public LoginPageExtended(WebDriver driver) {

    super(driver);

  }

  private String getErrorText(String name) {

    var elements = driver.findElements(By.tagName(name));

    if (elements.size() == 0)

      return "";

    return elements.get(0).getText();

  }

  public String getUsernameError() {

    return getErrorText("h3");

    // var usname = driver.findElement(By.className("error"));

    // return usname.getText();

  }

  public String getPasswordError() {

    return getErrorText("h3");

  }

  public LoginPageExtended setUsername(String username) {

    driver.findElement(By.id("user-name")).sendKeys(username);

    return this;

  }

  public LoginPageExtended setPassword(String password) {

    driver.findElement(By.id("password")).sendKeys(password);

    return this;

  }

  public String getInvalidCredentialsError() {

    return getErrorText("h3");

  }

}

**LoginTestsExtended**

package com.planittesting.swaglabs.tests;

import static org.junit.jupiter.api.Assertions.assertEquals;

//import com.planittesting.swaglabs.model.pages.LoginPage;

import com.planittesting.swaglabs.model.pages.LoginPageExtended;

import org.junit.jupiter.api.Test;

public class LoginExtendedTests extends BaseTestSuite {

  @Test

  public void verifyLoginWithInvalidCredentials() {

    var loginPageExtended = new LoginPageExtended(driver)

        .setUsername("username")

        .setPassword("password")

        .clickLoginButton();

    assertEquals("Epic sadface: Username and password do not match any user in this service",

        loginPageExtended.getInvalidCredentialsError());

  }

  @Test

  public void verifyEmptyUsernameError() {

    var loginPageExtended = new LoginPageExtended(driver)

        .clickLoginButton();

    assertEquals(

        "Epic sadface: Username is required",

        loginPageExtended.getUsernameError());

  }

  @Test

  public void verifyEmptyPasswordError() {

    var loginPageExtended = new LoginPageExtended(driver)

        .setUsername("standard\_user")

        .clickLoginButton();

    assertEquals(

        "Epic sadface: Password is required",

        loginPageExtended.getPasswordError());

  }

}